GENETIC ALGORITHMS FOR A SINGLE-TRACK VEHICLE AUTONOMOUS PILOT

Dana Vrajitoru
Intelligent Systems Laboratory
Computer and Information Sciences
Indiana University South Bend
South Bend, IN 46634, USA
email: danav@cs.iusb.edu

Poornima Konnanur
Intelligent Systems Laboratory
Computer and Information Sciences
Indiana University South Bend
South Bend, IN 46634, USA
email: npoornima@yahoo.com

Russel Mehler
Intelligent Systems Laboratory
Computer and Information Sciences
Indiana University South Bend
South Bend, IN 46634, USA
email: rmehler@cs.iusb.edu

Abstract

In this paper we present an application of genetic algorithms to an autonomous pilot designed for motorized single-track vehicles (motorcycles). The pilot is implemented as a multi-agent application using a physical model of the motorcycle and is embedded in an interactive application. We compare the performance of the configuration obtained by genetic algorithms with the manual configuration of the pilot and with the performance of human players.

The main contribution of the paper is proposing a model for piloting a single-track vehicle based only on perceptual information such as it would be observed by a human in the case of a real vehicle, and showing how the genetic algorithms can contribute efficiently to configuring the autonomous pilot.

Key Words
Genetic algorithms, multi-agents, autonomous pilot.

1 Introduction

Single track vehicles (STV) present somewhat different challenges than double-track vehicles like cars because their balance is not automatically achieved by the disposition of the wheels, but is a dynamic result of their motion. The input from the rider also comprises an additional component as the centrifuge force can be used to change direction.

The paper contains two main contributions: first, showing how a single-track vehicle can be driven on a road based on perceptual information in the absence of a specified path, and second, showing how the genetic algorithms can be used to configure the autonomous pilot faster and more efficiently than manually.

The autonomous pilots are an important aspect of developing the vehicles of the future and they represent an interesting challenge for intelligent control applications as well as for traffic control [12, 18]. This project starts from a simulation of a vehicle with a multi-agent autonomous pilot using perceptual information. The application aims to control the vehicle in a non-deterministic way inspired from the behavior of a human driver and using similar perceptual information to make decisions.

This paper presents an application of genetic algorithms to configuring an autonomous pilot for STVs. The pilot, introduced in [20], is implemented as a multi-agent model using perceptual information. Previously, the pilot was configured manually by trial and error, method that achieved a reasonably good performance, but was imprecise and time-consuming. The current research aims to improve the process, making it faster and with a better performance. We believe that the genetic algorithms are a good choice for this problem since we are faced with an optimization on multiple variables that is difficult to achieve by hill-climbing.

The genetic algorithms (GAs) [10, 9] as well as other evolutionary approaches, are a frequent choice for optimization problems with many applications. They have been successfully applied to related areas such as path-find for robots [3, 19], scheduling [2], and behavior development [13]. Several approaches have applied multi-agent models to the simulation of autonomous drivers [15] and our application follows similar ideas.

Most of the research on autonomous pilots is directed toward piloting aircrafts [14, 16, 1, 6], and cars [17]. Our approach targets motorcycles which have not been studied yet as extensively as the other types of vehicles and which represent a more challenging modeling problem.

Many of the autonomous pilot simulations, as for example, [4], propose an approach to driving a vehicle based on an outline of the direction of movement that the pilot must follow. While this approach is valid in a simulated environment, a system capable of driving a real vehicle would have to deal with situations where the precise path is not known. In this paper we present an approach where the shape of the road is not known, but must be inferred from perceptual cues that the pilot senses while driving the vehicle. Some similar studies exist for robots that move in an environment while trying to avoid obstacles and aim to reach a given target. Our model is different from those in general settings of the problem, which consist in driving a vehicle within a given road, following the general forward direction on this road without a given path or sequence of precise points to reach. We aim to simulate a real environment where a vehicle must be driven along a track based only on the information that can be obtained from visual sensors.

The autonomous pilot is a multi-agent probabilistic application where each agent is an independent process acting on one of the control units of the vehicle, as for example, the gas, the brakes, the handlebars, or the steering wheel. The agents use some information about the current status of the vehicle to make a decision about an action to be taken on their respective control units. This information includes both status data, like the current speed, and perceptual data, like the visible distance on the road in the direction of movement, the lateral distance to the border of the road, and the current slope. The performance of the

automatic pilot is compared with the performance of a trained human.

The paper is structured the following way. Section 2 describes the physical model and the equations that we have used for our simulation. Section 3 introduces our multi-agent automatic pilot. Section 5 presents the details of the application of genetic algorithms to this problem and the experimental results of this method. The paper ends with some conclusions.

2 Physical Model of the Single-Track Vehicle

In this section we introduce the physical model of a single-track motorized vehicle, as for example a motorcycle, and the equations of motion we used to implement the interactive application.

2.1 The Vehicle Control and Degrees of Freedom

A motorized STV, as for example, a motorcycle, can be modeled as a system composed of several elements with various degrees of freedom that can be driven through several control units. Figure 1 shows the components of the physical model of a motorcycle, where b is the distance between the wheels.

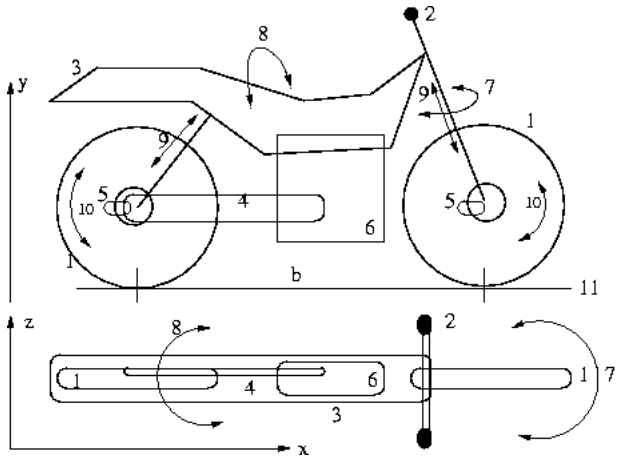

Figure 1. A motorcycle with control units and degrees of freedom: 1 - wheels, 2 - handlebar, 3 - saddle, 4 - chain, 5 - brakes, 6 - engine, 7 - steering, 8 - leaning, 9 - suspension, 10 - wheel spin, 11 - contact line

An STV is a non-holonomic dynamic system with six degrees of freedom: the rotation of the wheels around an axis parallel to Oz, the rotation of the handlebar and of the front wheel around the fork axis (steering), the front and back translation along the suspension axis, and the rotation of the whole vehicle around the Ox axis in a system of coordinates relative to the motorcycle where the origin is in the center of the vehicle, on the ground level.

The driver can control the vehicle through five inputs: the handlebar steering, leaning the vehicle laterally, the throttle, and the two brakes, front and back.

The state of the STV is described at any moment by the current position of the vehicle's center on the road, and by the current direction of movement which can be described either as a vector or as an angle in the (x, z) plane plus a slope, in general determined by the road. The model must also include the state of each unit of the STV with a degree of freedom and the current action of the driver on each of the control units. These two components are in general defined relatively to the STV's internal system of reference.

Various aspects of the physical model of the bicycle have been studied before. The closest model to our purposes is [7] which considers a bicycle as a nonlinear, nonholonomic, non-minimum phase system.

The stability and control of a bicycle are also of interest [11], [8], as well as the study of its aerodynamics [5].

2.2 STV Motion and Control

The STV is modeled as a reduced state system of continuous variables. The generalized coordinates of the vehicle at a particular moment are given by

$$q = (s, \quad , \theta)^T \tag{1}$$

where $s(t) = (x(t), z(t))$ represents the *spatial position* of the STV, θ is the *orientation angle* determining the *direction of movement* $d = (\cos\theta, \sin\theta)$, and the *leaning angle*. Let ϕ be the *steering angle*. Since our STV is based on a motorcycle and not on a bicycle, we have imposed the constraint that $-\pi/3 \leq \phi \leq \pi/3$. Figure 2 illustrates these angles and coordinates.

The vertical component of both s and d is determined by the road altitude and slope given the spatial position and orientation of the vehicle. In this paper we consider the road to be close enough to the sea level such that the gravitational acceleration is the constant g and the altitude of the vehicle does not really influence its motion. Let $\sigma(s, d)$ be the angle made by the contact line of the vehicle with the (x, z) plane, depending on its position and orientation.

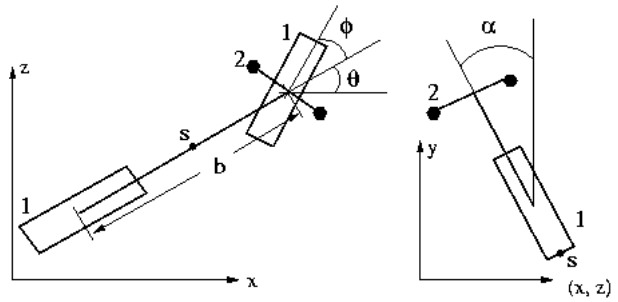

Figure 2. STV coordinate system: 1 - wheels, 2 - handlebar

The driver's input into the system is defined by the tuple $u = (\tau, \quad_f, \quad_r, \phi, \quad)$ where τ is an acceleration component along the moving direction d and $\quad_f$, $\quad_r$ represent the front and rear brakes respectively.

The nonholonomic constraints can be expressed by the following, where b is the distance between the two contact points of the wheels on the ground:

$$-x' \sin\theta + z' \cos\theta = 0 \tag{2}$$

$$b \cos\phi\, \theta' - \sin(\phi + \theta)x' + \cos(\phi + \theta)z' = 0 \tag{3}$$

Equation 2 expresses the fact that the STV moves in the direction of the vector d. Equation 3 allows us to compute the change in orientation due to steering. Both equations are adapted from [7].

In particular, if $-\pi/2 < \phi < \pi/2$, we can compute the change in the orientation angle due to steering as

$$\Delta\theta = \frac{\sin(\phi + \theta)\Delta x - \cos(\phi + \theta)\Delta z}{b \cos\phi} \tag{4}$$

Let $v = s'$ be the momentary speed or velocity in the direction of movement, and $a = v' = s''$ the momentary acceleration in the direction of movement. We consider the motion of the vehicle defined by

Newtonian mechanics. The position and velocity of the vehicle at $t + \Delta t$ are defined by $s(t + \Delta t) = s(t) + \Delta s$, $v(t + \Delta t) = v(t) + \Delta v$ where:

$$\Delta s = d \quad v \cdot \Delta t + a\frac{\Delta t^2}{2}\Bigg)\Bigg(\quad \Delta v = a \cdot \Delta t \tag{5}$$

In our case, the acceleration is defined by the gravity, the friction, the drag, and the throttle. The brakes do not act as a simple negative acceleration, but their input contributes to the drag and friction forces slowing the vehicle down.

To compute the drag force, a drag coefficient due to the brakes is added to the air resistance which is rather small, and to the engine brake which prevents the speed of the vehicle from increasing indefinitely. The resulting drag coefficient D is given by the following equation:

$$D = k_a + k_d(B_f + B_r) + k_e \tag{6}$$

where B_f and B_r represent the brakes input, k_a is the air resistance, k_d is the drag coefficient of the brakes, and k_e is a coefficient due to the engine brake. These forces cause the speed of the vehicle to become constant after a while for any given throttle input τ, and also cause the vehicle to eventually stop when no throttle input is provided anymore. Together with the friction force, they will prevent a resting motorcycle from going downhill if the slope σ is not null, and also prevent the speed from increasing indefinitely due to a gravitation in the direction of movement when the vehicle is going downhill.

The resulting acceleration is defined by

$$a = \tau + g\ \sin\sigma - k\ g\ \cos\sigma - k_b(B_f + B_r) - Dv^2 \tag{7}$$

where $g = 9.8\ m/s^2$ is the gravitational acceleration at sea level, k is the coefficient of friction, and k_b is a coefficient of friction for the brakes.

The latest model of the motorcycle improved from [20], implements an additional change in the direction of movement due to leaning that is equally available for the human player, together with fail-safe conditions that can cause the vehicle to crash if they are not met, as for example, leaning too far for the current velocity.

2.3 Leaning

We can define the leaning problem the following way: given the leaning angle and the speed v, what is the radius of the circle on which the STV rotates around Oy?

The first force that we are going to consider that affects the change in direction of the vehicle due to leaning is the *centrifuge force*, illustrated by Figure 3 and defined by

$$F_c = m\ \omega^2\ r \tag{8}$$

where ω is the angular speed, and r is the radius of the circle on which the object is turning. If v is the horizontal speed, then we can define the angular speed as $\omega = v/r$, so the centrifuge force is equal to

$$F_c = m\frac{v^2}{r} \tag{9}$$

A second force that interacts with the vehicle in the lateral movement is the lift due to the friction with the air. We can adapt an equation taken from airplane wing simulation that computes the *lifting force* F_L as

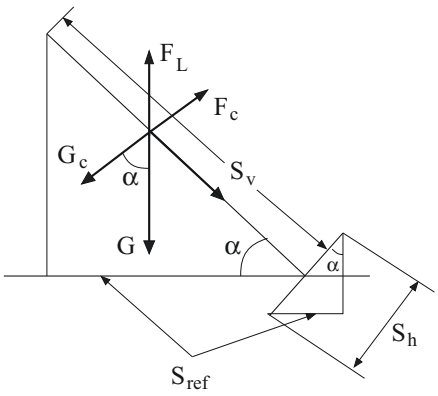

Figure 3. The forces and quantities involved in leaning

$$F_L = \frac{1}{2}\rho v^2 S_{ref} C_L \tag{10}$$

In this equation ρ is the air density, that we can consider to be approximately $\rho = 1.22145 kg/m^3$ at 0 altitude. S_{ref} is the reference area, the horizontal projection of the vehicle surface.

The last component of the lateral movement is the gravitational force itself, which has a norm equal to $g\ m$. From this force, we subtract the lifting force first. Starting from the same angle α, the resulting gravitation force, which is vertical, is decomposed in a force along the vertical axis of the motorcycle and another one that is normal to the motorcycle. The rotation will be determined by the component that is perpendicular to the motorcycle axis. This component, that we call *central gravitation* and denote by G_c, is given by

$$G_c = (g\ m - F_L) \sin \alpha \tag{11}$$

By imposing the condition that the central gravitational force should be equal to the centrifuge force, we can compute the rotation radius r:

$$r = \frac{m\ v^2}{(g\ m - F_L) \sin \alpha} \tag{12}$$

3 The Autonomous Pilot

In this section we present the ideas that we used to implement the autonomous pilot for our motorcycle.

Our autonomous pilot is composed of several agents for the various control units of the vehicle. We chose a multi-agent model because it allows us to define the equations and parameters for each of the control units independently. Combining them into a single pilot would introduce an unnecessary complication in the equations and would render the configuration process more difficult.

Each of the agents composing the pilot acts independently in adjusting their respective control units. They are implemented as fuzzy controllers, the precise equations for each of them being specified further down in this section. For each of them, a series of conditions for action are specified, a particular course of action being prescribed in each case. The resulting action taken by the agent is obtained by balancing or averaging on the actions resulting from each condition.

A small part of the communication between the agents is achieved through *events* that they generate, such as the update event being generated by the alert agent. The most part of the communication is

implicit, through the changes generated to the state of the system and through the internal and external perceptual cues. For example, if the brake agent takes action in one state of the system to reduce the speed, a slower velocity in the next frame will cause the gas agent not to decrease the gas to adjust to a higher turn in the road. Our model is similar to an ant colony in this respect, where the individual ants do not communicate explicitly, but implicitly through the changes made to their environment, namely the presence of pheromones.

3.1 Perceptual Information

The autonomous pilot is using perceptual information to make decisions about the vehicle driving. This information is inspired from the perceptual cues that a human driver would also be paying attention to while driving a vehicle.

In our application, the pilot is aware of the following measures:

The *visible front distance*, denoted by $front$, defined as the distance to the border of the road from the current position in the direction of movement, scaled by the length of the vehicle. This distance is a measure of how much of the road is visible ahead and also of how straight the road is in front of the vehicle. We will also mention this as the *horizon*.

The *front probes*, denoted by $frontl$ and $frontr$, are defined as the distances to the border of the road from the current position of the vehicle in directions rotated left and right by a small angle from the direction of movement. They give the pilot an indication as to which way from the direction of movement the front distance would become larger, which can be used to turn that way.

The *lateral distances*, denoted by $leftd$ and $rightd$, are measures of the lateral distance from the vehicle to the border of the road, at a short distance in front, simulating what the pilot might be aware of without turning their head to look. In our computations we use $lat_n = |\frac{leftd-rightd}{\max(leftd,rightd)}|$, the normalized difference between the lateral distances. A high value of this measure indicates a turn in the road, or that the vehicle is close to the border. A value close to 0 indicates that the vehicle is in the middle of the road.

The $slope$, is a perceptual version of σ, discretized to simulate the intuitive notion of road inclination that a human driver would have, approximated by the fuzzy values almost flat, slightly inclined up or down, or highly inclined up or down. This simulates the fact that a human pilot is not aware of the precise value of σ.

Figure 4 shows an example of the geometrical definition of these measures.

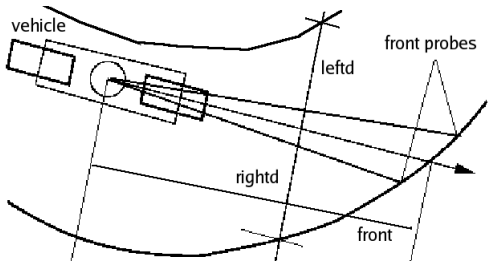

Figure 4. Perceptual information used by the autonomous pilot

Beside the perceptual information, the autonomous pilot uses the current status of the motorcycle to make decisions about the action to be taken on each of the control units of the vehicle. The status includes measures like the v, τ, B_f, B_r, ϕ.

3.2 Control Units

The motorcycle is driven by several control units (CUs). Each of them is controlled by an independent agent with a probabilistic behavior. The agents are not active during the computation of each new frame simulating the evolution of the vehicle on the road, but only once in a while in a non-deterministic manner. This simulates the behavior or a human driver that may not be able to respond instantly to the road situation and allows for a certain reaction time.

The current control units focus on the gas (throttle), the brakes, the handlebar/leaning. Each of these CUs is independently adjusted by an agent whose behavior is intended to drive the motorcycle safely in the middle of the road at a speed close to a given limit. In our case, the agents controlling the throttle and the handlebar are in general more active than the agent controlling the brakes.

The next paragraphs introduce the equations used by each of our agents to make a decision and to perform an action. The equations comprise a fair number of coefficients and thresholds. The configuration of each agent uses independent values for the coefficients.

The Throttle. This agent controls the amount of gas supplied to the engine and influences the acceleration that the vehicle is submitted to.

The input for this agent is represented by $(v, front, leftd, rightd, slope)$. The agent uses a minimal speed threshold v_{low}, a maximal speed threshold over which the speed is considered unsafe, and the given speed limit v_{limit}. The agent aims to keep the vehicle speed above v_{low} and below the maximal one, and also close to the v_{limit}.

The agent detects a turn in the road by testing lat_n and eventually decreases τ to allow for a safe turn. A similar rule is applied to the visible distance in front of the driver: a low value for $front$ indicates an unsafe road situation requiring a reduced τ. In any other situation it attempts to keep the speed close to v_{limit}.

Equation 13 represents the conditions under which τ should be decreased, causing the vehicle to slow down. If the condition bellow i not fulfilled, τ is given an appropriate value to keep the speed close to the limit.

$$v > v_{low} \wedge$$
$$(v > v_{limit} \ \vee \ tr_{lat} > lat_n \ \vee \ tr_{fr} > front) \tag{13}$$

Equation 14 illustrates the change in throttle performed by the agent, where c_{incv}, c_{decv}, and c_{sl} are configurable coefficients. The actual amount of the change is a probabilistic quantity equally distributed in a small neighborhood around the computed value, to account for the human imperfection.

$$\Delta\tau = c_{incv}(front - thr_{fr})(v - v_{low}) +$$
$$c_{decv}\left((v - v_{limit}) + tr_{lat} + tr_{fr}\right) + c_{sl} \cdot slope \tag{14}$$

The Brakes. The agent controlling the brakes presents a similar behavior to the throttle agent because the rules according to which the speed should decrease are of general purpose. The equations of this agent are simpler because the brakes can only decrease the speed and not increase it. The speed is only decreased when a more drastic change is necessary than we can assume will be achieved only by decreasing τ.

Equation 13 is also used to determine when to apply a force on the brakes, but the coefficients c_{incv}, c_{decv}, and c_{sl} can have different values for this agent. The change in either B_r or B_f is based on the following equation.

$$\Delta B_{r,f} = c_{decv}\left((v - v_{limit}) + tr_{lat} + tr_{fr}\right) - c_{sl} \cdot slope \tag{15}$$

The Steering / Leaning Agent. The motorcycle can achieve a change in direction either by steering using the handlebar, or by leaning. The autonomous pilot has been tested under three conditions: when

the motorcycle is driven entirely by steering, when the motorcycle is driven entirely by leaning, and a combined strategy, consisting of steering at low speed (bellow a threshold) and leaning at a speed above the threshold. This emulates the general strategy employed by human drivers.

The agent controlling the handlebar and leaning of the motorcycle is the one with the most complex behavior. This agent is also using the lateral distances to the border of the road, as well as the front probes $frontl$ and $frontr$, to make decisions about turning left or right. The agent turns the vehicle in the direction of the longer distance between the left and right, getting away from the closest border.

The agent first considers the immediate distance to either side, given by the lateral distances. Thus, if the vehicle is not situated within a given percentage (like 20%) of the center of the road, then the agent moves the vehicle towards the center.

If the first measure (lateral distances) does not provide a condition for the vehicle to turn, the agent estimates the distance forward to the horizon. Based on the front probes and the front distance, the agent moves towards the center of the horizon (given by the front probes). The handlebar and leaning angles depend on the distance to the horizon, a bigger change being required if the horizon is closer.

The agent starts by making a decision whether to use the lateral distances as reference or the front probes. Let us denote by $probe_n$ the normalized difference between the front left and right probes as shown in Equation 16 and by $probe_{abs} = |probe_n|$ the absolute value of this quantity.

$$probe_n = \frac{frontl - frontd}{\max(frontld, frontd)} \tag{16}$$

Let us denote by lat_{diff} the quantity used by the agent to decide if it must turn and in which direction, computed according to the following equation.

$$lat_{diff} = \begin{cases} lat_n & \text{if} \quad lat_n > thr_1 \text{ and } frontd > thr_2 \\ \frac{lat_n + probe_n}{2} & \text{if} \quad lat_n > thr_3 \text{ and } frontd > thr_4 \\ probe_n & otherwise \end{cases} \tag{17}$$

where $thr_i, i = 1, 4$ are configurable coefficients.

The amount of the angle change depends on the measure lat_{diff} and on the speed. Thus, if the speed of the motorcycle is lower, the turning angle must be higher to achieve a given change in direction. If the vehicle moves at a higher speed, smaller angle changes are necessary to obtain the same change in direction, both by the handlebar angle or by using the leaning angle.

The agent will update the handlebar angle or the leaning angle if the condition expressed in Equation 18 is fulfilled. This means that a change is necessary either if the lateral difference measure is greater than the threshold thr_{lat} (the vehicle is off-center), or if the distance in the direction of movement to the border of the road is smaller than another threshold, thr_{front}, meaning that the vehicle might get off the road soon if it continues in the current direction.

$$|lat_{diff}| > thr_{lat} \quad \text{or} \quad front < thr_{fr} \tag{18}$$

If we denote by $\Delta\phi = \phi(t + \Delta t) - \phi(t)$, then the general rule for modifying the orientation of the handlebar is shown in Equation 19, but the actual amount of the change is a probabilistic quantity equally distributed in a small neighborhood around the computed value.

$$\Delta\phi = c_{hbar} \quad lat_{diff} + \frac{thr_{fr} - front}{thr_{fr}} \Bigg) \Bigg(\tag{19}$$

The change in the leaning angle is computed based on a similar equation as for $\Delta\phi$, but the coefficients can be configured independently.

Alerting Agent. Beside all the agents that are in direct control of the motorcycle, the pilot comprises a fourth agent that does not perform any action on the vehicle. While the other agents are active only occasionally, this agent is probing the vehicle and road conditions for every new frame and is capable of activating one of the other agents if the situation requires special attention. Thus, if the speed of the vehicle is either too high or too low, if the visible front distance is too short, or if the difference between the left and right lateral distances is too high, this agent considers the situation to be exceptional, meaning unsafe, and generates an alert event that will randomly activate one of the agents that can take action on the motorcycle and correct the issue.

Equation 20 describes the condition under which the alerting agent generates an update event that triggers one of the other agents. The alerting agent does not decide which other agent will perform the necessary action.

$$\begin{aligned} v < \ c_{vlow} v_{limit} \ \vee \ v > \ c_{vhigh} v_{limit} \ \vee \\ lat_{abs} < \ thr_{lat} \ \vee \ front < \ thr_{fr} \end{aligned} \tag{20}$$

4 Implementation

The program is implemented in C++ using OpenGL. The source code with compilation instructions is available in the software section of the web site of the Intelligent Systems Laboratory of Indiana University South Bend. `http://www.cs.iusb.edu/~isl/` The project is composed of four parts:

- the graphics engine,
- the physics engine of the motorcycle,
- the autonomous pilot,
- the user interface.

Each of the four modules is implemented as a set of self-contained classes and units with constrained interaction with other units, following a modular model that is easier to update and that could be easily adapted to a different piloting system in the future. For example, to apply the genetic algorithm to configuring the pilot, we disabled the graphic implementation of the system and the user interface. The program structure made this kind of change seamless.

Graphics Engine

The graphics engine implements the scene objects and their conversion into OpenGL commands. It contains separate classes for the geometry, the animation, and the behavior of the motorcycle, the pilot, and the road. The geometry classes define the geometric objects composing the motorcycle and of the pilot. The animation classes define the basic animation properties of these objects, starting with the definition of scene graphs providing the internal structure of the motorcycle and the pilot. The behavior classes provide higher level functionality for the scene graphs.

For example, the geometric objects composing the wheels are described in the motorcycle geometry class. The relative position of each of them as a subtree in the motorcycle tree, as well as the internal and relative rotation of the wheels are defined in the motorcycle animation class. The transformations applied at each new frame to apply the current position, orientation, and speed are defined in the behavior class of the motorcycle.

Physics Engine

The physics engine of the motorcycle contains the functionality of the vehicle implementing the kinematics laws and its interactions with the environment. This component implements the physics aspects described in Section 2. Essentially, this model is responsible for computing the new state of the system from the current one taking into consideration the actions currently in process on each control unit of the motorcycle.

Autonomous Pilot

The autonomous pilot communicates with the graphics and physics agent through a limited number of actions to be taken on each control unit, as for example, increasing the gas by a given amount. A generic Agent class describes the basic functionality of all the agents as virtual functions, as for example, the callback mechanisms for event reaction. Each individual agent overloads the functionality of the base class to adapt it to its own behavior.

User Interface

The user interface provides visual display of the state of the system for every new frame and animation, as well as the interaction of the player with the vehicle. The main application window contains a mini-map that can be used to monitor the vehicle progression on the circuit. The perceptual information is also displayed as small spheres marking the intersection of the direction of movement with the road, or the lateral points of reference.

Figure 5 shows the application pipeline and the interaction between the components of the program. Figure 6 shows the main window of the application displaying the motorcycle on the road, featuring a mini-map in the top left corner and the perceptual information.

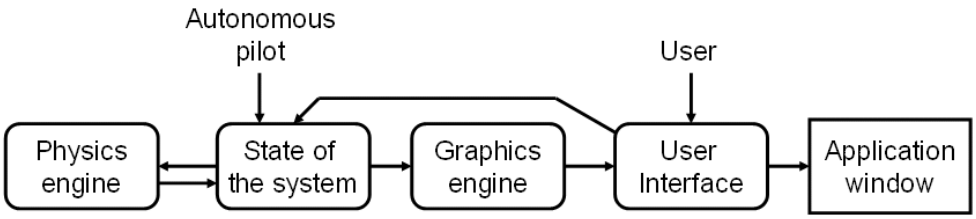

Figure 5. Program components and application pipeline

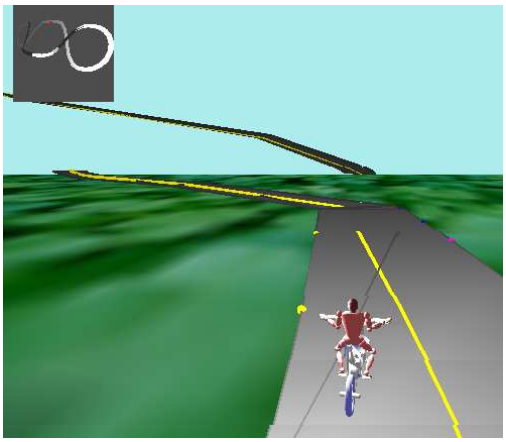

Figure 6. The main application window displaying the vehicle

5 Applying Genetic Algorithms to the Pilot Configuration

All of the agents composing the autonomous pilot are governed by equations such as we have introduced in the previous section. These equations comprise a set of thresholds and constants that can be configured to adjust its behavior and optimize its performance.

To apply the GAs to this problem, we have chosen a representation where each configurable coefficient is assigned 10 binary genes, and the chromosome results by concatenating all of these strings of genes. Thus, we worked with 32 coefficients for the leaning and steering modes, and with 36 for the combined mode, which means that the chromosome is of a length of 320 and 360 respectively.

We have chosen the one-point crossover for our experiments with a probability of 0.8 and a mutation rate of probability 0.01. We have used a monotonous reproduction where the best individual from the old generation replaces the worst from the new one if there is a decay in performance in the new generation.

A chromosome was evaluated by running the motorcycle in a non-graphical environment once with the pilot configured based on values obtained by decoding the chromosome over the same circuit as the manually configured pilot. For this we had a number of 50 reference points on the road which were marked when the motorcycle passed next to them. The fitness was computed as follows:

$$F(x) = \begin{cases} \frac{d_m}{d_t} + \frac{1}{1+t_m} & \text{if the circuit was completed} \\ \frac{d_m}{d_t} + \frac{1}{5+t_m} & \text{if the circuit was not completed} \end{cases} \tag{21}$$

where d_m is the number of points crossed by the motorcycle, d_t is the total number of points, and t_m is the total time taken until either the circuit was completed, or until a failure condition was detected.

Thus the fitness reflects both how much of the circuit the motorcycle completed, and how fast it was capable of finishing the track. In general, a fitness higher than 1 is an indication of completion of the circuit.

A failed circuit can be caused by one of the following three situations: a crash occurred because of too much leaning, the motorcycle exited the road without returning soon enough, or the motorcycle reached the starting point again without touching all the intermediate points, which means that it took a shortcut somewhere.

5.1 Experiments

The pilot was tested using a circuit consisting of 3 loops such that a portion of it is elevated to test the gravitational effects. The circuit was designed with the intention to test the ability of the pilot to drive correctly in situations where the road is turning both to the left and to the right, and also where the slope of the road is going uphill or downhill. The track is shown in Figure 7.

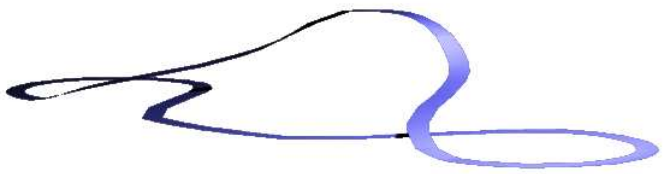

Figure 7. Test circuit for the experiments

We have performed 100 runs of the GAs for each of the 3 modes, steering, leaning, and combined. We selected the chromosome with the highest fitness over all the trials and performed 100 trials on the

Table 1. Performance of the human players

	Human 1	Human 2
Total time	97.4	79.2
Average speed	6.19	8.94
Maximum speed	8.75	12.26
Total distance	2312.05	2316.83
Lateral balance	0.29	0.36
Completed circuits	100%	100%

circuit with a pilot configured based on it, in a graphical environment this time. We compared these new results with 100 trials of the manually configured pilot in the same conditions, as presented in [20].

The results of these experiments are presented as follows. Figure 8 shows the evolution of the fitness in the three modes in 100 generations. We can notice from it a big difference in performance between the three modes. In steer mode, the fitness suggests that the derived pilot was in general capable of completing the circuit. In the lean mode the fitness indicates that most circuits were ended in failure. In the combined mode the fitness achieved a value close to 1, which means that the pilot probably succeeded in completing the circuits at least some of the time.

Table 1 introduces some statistics showing the performance of two human players, each having completed the circuit 5 times. The following statistics were taken into consideration: the average time taken to complete the circuit, average speed over the entire circuit, maximal speed that the player has achieved at any time, and total distance covered to complete the circuit, which is a measure of efficiency. The next measure is the average value of lat_{abs}, marked by lateral balance, taking values between 0 and 1. 0 indicates the center of the road, 1 the extreme borders of the road, and lower values indicate a better driver behavior on the road. The last line shows the percentage of completed circuits in each case.

Table 2 shows the performance of the manually configured autonomous pilot in 100 trials. The columns entitled **C** and **I** mark the statistics made on completed versus incomplete circuits respectively. Thus, the pilot with this configuration has completed the circuit 92 times out of 100 in the steering mode, and was not capable of completing the circuit in the leaning and combined modes. The average speed was in general much lower than the human players.

Tables 3 and 4 show the performance of the pilot configured based on the parameters derived by the GA, for the completed and incomplete circuits respectively. From these tables we can see that the pilot was capable of completing at least one circuit in each mode. Moreover, in the steering mode each of the 100 circuits was completed successfully. The average speed is higher than in the manual configuration case, which is due to the fact that the fitness function favors the configurations that finish the circuit faster. Overall the performance is much closer to the human players.

A T-test with a 95% confidence on the average time and speed indicated a significant improvement of the configuration derived by GAs over the manually configured pilot in each of the test cases.

6 Conclusions

In this paper we presented an application of the genetic algorithms to configure a multi-agent autonomous pilot for motorcycles. The application is implemented based on the physical equations describing the vehi-

Table 2. Results of the manually configured pilot

	Steer		Lean	Combined
	C	I	I	I
Total time	327.75	78.25	15.79	25.58
Total distance	2338.77	720.42	77.23	108.081
Speed	1.96	2.58	1.25	1.46
Max speed	4.99	4.96	2.56	2.82
Lateral balance	0.32	0.33	0.55	0.44
Completed circuits	92%		0%	0%

Table 3. Results of the GA configured pilot, completed circuits

	Steer	Lean	Combined
Total time	133.73	221	508.67
Total distance	2333.33	2356.08	2361.24
Speed	4.16	2.74	1.01
Max speed	6.49	6.08	2.59
Lateral balance	0.40	0.40	0.30
Completed circuits	100%	1%	3%

Table 4. Results of the GA configured pilot, incomplete circuits

	Steer	Lean	Combined
Total time	n/a	52.60	105.2
Total distance	n/a	522.46	495.96
Speed	n/a	2.07	0.91
Max speed	n/a	5.09	1.87
Lateral balance	n/a	0.46	0.36
Incomplete circuits	0%	99%	97%

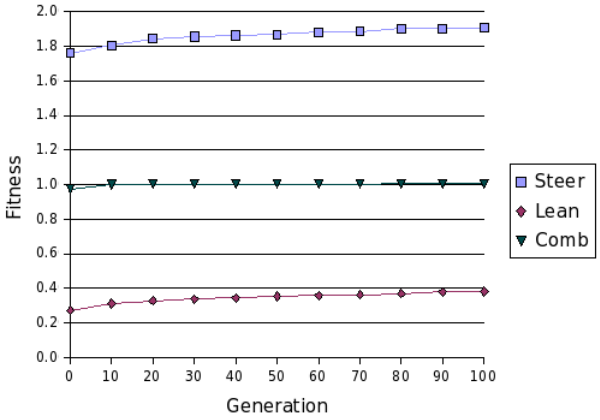

Figure 8. Best fitness in 100 generations

cle's attributes, motion, and road behavior. The physical model of the vehicle was described in Section 2.

The autonomous pilot is composed of several agents, as described in Section 3, each of them being defined by a set of equations with configurable parameters. Our previous work has focused on manually finding the best values of the parameters, which is a time-consuming and imprecise operation. The main focus of the research presented in this paper was on applying the genetic algorithms to configure the parameters used by the autonomous pilot.

The experiments described in Section 5 show that the pilot configured by genetic algorithms perform significantly better than the manually configured one. The number of completed circuits has increased, as well as the efficiency of the pilot in finishing the circuit. The pilot comes closer to the human performance with the same application. We can entail that the genetic algorithms are indeed a good choice for this particular task.

As a direction for future research, the equations governing the behavior of the pilot could be improved for the leaning mode, where the number of completed circuits is still low even after applying the genetic algorithms. A possible approach to this problem would be to apply genetic programming. So far the model was only applied to a simulated vehicle. Although applying it to an actual vehicle presents many technical difficulties, we envision a possibility of development by transposing the driving model to an actual robot or a toy vehicle.

References

[1] T. F. Abdelzaher, E. M. Atkins, and K. G. Shin. QoS negotiation in real-time systems and its application to automated flight control. *IEEE Transactions on Computers*, 49(11):1170–1183, 2000. Best of RTAS '97 Special Issue.

[2] D. Abramson, G. Mills, and S. Perkins. Parallelisation of a genetic algorithm for the computation of efficient train schedules. In *Proceedings of the Parallel Computing and Transputers Conference*, pages 139–149, 1993.

[3] R. R. Cazangi, F. J. Von Zuben, and M. F. Figueiredo. Autonomous navigation system applied to collective robots with ant-inspired communication. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 121–128, Washington, D.C., 2005. ACM Press.

[4] B. Chaperot and C. Fyfe. Improving articial intelligence in a motocross game. In *Proceeding of the IEEE Symposium on Computational Intelligence and Games*, Reno, USA, 2006.

[5] A. Fuchs. Trim of aerodynamically faired single-track vehicle in crosswinds. In *Proceedings of the 3rd European Seminar on Velomobiles*, Roskilde, Denmark, 1998.

[6] V. Gavrilets, E. Frazzoli, B. Mettler, M. Piedmonte, and E. Feron. Aggressive maneuvering of small helicopters: a human centered approach. *International Journal on Robotics Research*, 2001.

[7] N. Getz. Control of balance for a nonlinear nonholonomic no-minimum phase model of a bicycle. In *American Control Conference*, Baltimore, June 1994.

[8] N. Getz and J. Marsden. Control for an autonomous bicycle. In *IEEE International Conference on Robotics and Automation*, Nagoya, Japan, May 21-27 1995.

[9] D. E. Goldberg. *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley, Reading (MA), 1989.

[10] J. H. Holland. *Adaptation in Natural and Artificial Systems*. University of Michigan Press, Ann Arbor, 1975.

[11] D. Jones. The stability of the bicycle. *Physics Today*, 23(4):34–40, 1970.

[12] T. Kelly. Driver strategy and traffic system performance. *Physica A*, 235:407–417, 1997.

[13] K. J. Lee and B. T. Zhang. Learning robot behaviors by evolving genetic programs. *Proceedings of the 26th International Conference on Industrial Electronics, Control and Instrumentation (IECON-2000)*, pages 2867–2872, 2000.

[14] I. Martinos, T. Schouwenaars, J. De Mot, and E. Feron. Hierarchical cooperative multi-agent navigation using mathematical programming. In *Proceedings of the Allerton Conference on Communication, Control and Computing*, 2003.

[15] T. Al-Shihabi and R.R. Mourant. Toward more realistic behavior models for autonomous vehicles in driving simulators. *Transportation Research Record*, (1843):41–49, 2003.

[16] J. De Mot and E. Feron. Spatial distribution of two-agent clusters for efficient navigation. In *IEEE Conference on Decision and Control*, Maui, HI, 2003.

[17] R.R. Mourant and S. Marangos. A virtual environments editor for driving scenes. In *Proceedings of the International Conference on Computer, Communication and Control Technologies*, volume IV, pages 379–384, Orlando, Florida, 2003.

[18] R.R. Mourant and D. Refsland. Developing a 3d sound environment for a driving simulator. In *Proceedings of the Ninth International Conference on Virtual Systems and Multimedia*, pages 711–719, Montreal, Canada, 2003.

[19] E. Pires, J. Machado, and P. Oliveira. Robot trajectory planner using multi-objective genetic algorithm. In *Proceedings of the Genetic and Evolutionary Computation Conference*, Washington, D.C., 2004.

[20] D. Vrajitoru and R. Mehler. Multi-agent autonomous pilot for single-track vehicles. In *Proceedings of the IASTED Conference on Modeling and Simulation*, Oranjestad, Aruba, 2005.